

# 基于资源的协作过滤推荐算法研究

纪良浩, 王国胤

Ji Liang-hao, Wang Guo-yin

重庆邮电大学 计算机科学与技术研究所, 重庆 400065

Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

E-mail: jilh@cqupt.edu.cn

Ji Liang-hao, Wang Guo-yin. Collaborative filtering based on item content. Computer Engineering and Applications, 2008, 44(8): 164-168.

**Abstract:** Collaborative filtering is the most prevalent algorithm of personalized service, but there are always two difficult problems for collaborative filtering algorithms, that is, data sparsity and expansibility. In this paper, a collaborative filtering algorithm is proposed based on item category. On the foundation of classifying items, it converts rating matrix of user-item into user-category of item. And then it clusters users, finds the nearest neighbors of active users in the sub-clustering that the active users exist in. Finally, it recommends to active users. Experiments show that the proposed algorithm reduces the data sparsity and dimensionality, its recommendations are good, and the simultaneity and expansibility of the algorithm is improved effectively.

**Key words:** collaborative filtering; sparsity; expansibility; clustering; recommendation

**摘要:** 协作过滤是当今应用最为普遍的个性化推荐算法, 然而数据的稀疏性和算法的可扩展性一直是协作过滤算法所面临的两大问题。提出了一种新的推荐算法——基于资源的协作过滤算法。该算法在对资源项目依内容划分的基础上, 将用户—项目评分矩阵转换为用户—资源类别评分矩阵, 然后对用户聚类, 在目标用户所在的簇中寻找其最近邻居并产生推荐。实验表明, 该算法不仅降低了数据的稀疏性和维度, 缩小了目标用户最近邻的查找范围, 算法的扩展性得到了有效改善, 而且提高了最近邻的准确度, 推荐精度较以往传统算法有明显提高。

**关键词:** 协作过滤; 稀疏; 扩展; 聚类; 推荐

**文章编号:** 1002-8331(2008)08-0164-05 **文献标识码:** A **中图分类号:** TP18

当前, Web 已成为人们获取信息和知识的一种不可或缺的手段, 然而随着 Web 信息的日益增加, 人们却不得不花费更多的时间来查找、浏览自己所需的信息。搜索引擎的出现, 满足了人们一定的需求, 却不能满足不同背景、不同目的和不同时期人们的查询请求, 从而导致了个性化服务技术的提出。个性化服务通过收集和分析用户信息来学习用户的兴趣和行为, 其能够更好地理解用户, 发现用户隐藏的兴趣和群体用户的行为规律, 从而制定相应的信息过滤策略, 按照用户的个性化信息进行主动的推荐服务。它能为不同用户提供不同的服务, 以满足各自不同的需求。运用个性化服务技术能充分提高站点的服务质量和访问效率, 从而吸引更多的访问者。个性化推荐可以采用基于规则的技术、基于内容过滤和协作过滤技术<sup>[1]</sup>, 其中运用最为普遍和成功的是协作过滤技术。

考虑到协作过滤技术所面临的数据高维稀疏和扩展性问题, 提出了一种新的基于资源的协作过滤推荐算法。实验表明,

较传统的协作过滤算法, 其不仅降低了数据的稀疏程度和维度, 提高了算法的扩展性, 同时在推荐精度上还有明显的提高。

## 1 相关工作

### 1.1 协作过滤概述

协作过滤(Collaborative Filtering)又称为合作过滤或社会过滤(Social Filtering), 它基于如下假设<sup>[2]</sup>: 如果一些用户对一些项目的评分较相似, 则他们对其它项目的评分也比较相似。协作过滤实现的基本思想是系统采用某种技术找到目标用户的若干邻居(与目标用户有相似兴趣的用户), 然后根据最近邻居对目标项目的评分产生推荐。

协作过滤分析的是“用户—用户”的关系, 根据用户兴趣的相似性来推荐资源, 用户对资源的评分向量作为用户兴趣的表现形式。用户对资源的评分可以是显式的也可以是隐式的。显式评价通常是用户以数值的形式对资源进行评分, 如果数值很

**基金项目:** 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60373111); 重庆市自然科学基金(the Natural Science Foundation of Chongqing City of China under Grant No.2005BA2003); 新世纪优秀人才支持计划; 重庆邮电大学自然科学基金。

**作者简介:** 纪良浩(1977-), 男, 讲师, 主要研究方向为智能信息处理, 个性化服务技术; 王国胤(1970-), 男, 博士, 教授, 博士生导师, 主要研究方向为人工智能, 智能信息处理等。

**收稿日期:** 2007-07-20 **修回日期:** 2007-11-06

高, 表示用户非常喜欢该产品, 反之表示用户不喜欢该产品。如果用户希望得到推荐系统的帮助, 首先需要向系统提交他对一些产品的评价信息。而隐式评价通常是从数据资源中派生出来的, 例如分析网站的日志文件等来分析用户的兴趣偏好并映射为显式的评价信息, 这种方法主要针对为未注册用户提供个性化服务。

## 1.2 数据源的数据结构

用户对资源的评分无论是采用显式还是隐式的形式, 最终都将映射为一个用户对资源的兴趣矩阵。假设有  $M$  个用户和  $N$  项资源, 则可以表示成一个  $M \times N$  的矩阵  $R$ , 其中每一项  $R_{m,n}=r$ , 表示用户  $m$  对资源  $n$  的评分为  $r$ , 如果  $R_{m,n}=\text{NULL}$ , 则代表用户  $m$  对资源  $n$  没有评分。表 1 为用户—资源项目评分矩阵  $R(M \times N)$ 。

表 1 用户—资源项目评分矩阵  $R(M \times N)$

Use	Item				
	Item <sub>1</sub>	...	Item <sub>j</sub>	...	Item <sub>N</sub>
User <sub>1</sub>	$R_{1,1}$	...	$R_{1,j}$	...	$R_{1,N}$
...	...	...	...	...	...
User <sub>i</sub>	$R_{i,1}$	...	$R_{i,j}$	...	$R_{i,N}$
...	...	...	...	...	...
User <sub>M</sub>	$R_{M,1}$	...	$R_{M,j}$	...	$R_{M,N}$

## 1.3 协作过滤算法

协作过滤算法可分为两大类<sup>[3]</sup>: 基于内存的算法(Memory-based)和基于模型的算法(Model-based)。

基于内存的算法利用整个用户—资源项目评分矩阵来产生推荐, 一般说来, 系统采用某种统计技术寻找目标用户的最近邻居, 这种方法也称为最近邻法或基于用户的算法(User-based)。

基于模型的算法一般采用机器学习的方法来构造一个用户的评分模型, 用概率的方法来预测目标用户对未评分项目的评分。Bayesian<sup>[4]</sup>网络技术、关联规则<sup>[5]</sup>技术及 Horting 图<sup>[6]</sup>技术等都属于基于模型的算法。

## 1.4 研究现状

随着用户和资源数目的不断增加, 协作过滤推荐技术所面临的一些问题也越来越凸现出来, 主要表现在以下两个方面:

(1) 数据的高维稀疏性。在资源项目很多的情况下, 用户的评分数据相对来说很少, 比如在大型的电子商务系统中, 用户的评分项目一般不会超过资源项目总数的 1%<sup>[3]</sup>。数据的高维稀疏性导致不同用户之间交叉评分的项目会很少, 这样用传统的用户相似性度量方法计算出来的用户最近邻居集就不准确, 从而导致信息推荐质量的下降和相似性计算耗费的增加。

(2) 算法的可扩展性。随着用户和资源项目数的增加, 系统的计算量会越来越大, 难以满足推荐系统适时性的要求。

对于数据高维稀疏性问题, 目前的解决方法主要有三种:

(1) 设置初始评分。将空缺的项目评分设定为一个固定的默认评分<sup>[2]</sup>或者基于项目的相似性预测用户对未评分项的评分<sup>[7,8]</sup>。设置初始评分的方法可以增加评分数据的稠密性, 提高最近邻生成的效率, 然而这种方法并不能从根本上解决数据的高维稀疏性问题<sup>[9]</sup>。

(2) 基于人工智能的方法。通过采用 Horting 图<sup>[6]</sup>、聚类<sup>[10-12]</sup>、贝叶斯网络<sup>[13]</sup>、软件代理<sup>[14,15]</sup>及粗糙集<sup>[16]</sup>等手段, 增加用户在项目空间上重叠的数目, 以降低数据稀疏性。但此方法在解决数据稀疏性问题的同时往往牺牲了推荐的精度, 并面临推荐计算

的可扩展性问题<sup>[17]</sup>。

(3) 基于降维思想的方法。该方法主要采用奇异值分解<sup>[18,19]</sup>、潜在语义索引<sup>[20]</sup>、主成分分析<sup>[21]</sup>、矩阵划分<sup>[22,23]</sup>等技术。通过将稀疏的用户项目矩阵转化为由主成分构成的稠密矩阵以解决数据稀疏问题。实验表明基于降维思想的方法显著提高了推荐系统的伸缩能力, 有效地解决了同义词问题。然而降维会导致信息的损失, 降维的效果与数据集密切相关, 并且在项目空间维数很高时, 降维的效果难以保证<sup>[24]</sup>。

对于扩展性问题, 研究者们发现基于模型的算法, 如贝叶斯网络模型<sup>[4]</sup>, 虽然可以在一定程度上解决算法的可扩展性问题, 但是该类算法比较适合用户兴趣爱好变化比较稳定的情况, 而且模型训练的代价一般比较大, 不太适合数据更新频繁的系统。文献[25]中提到了两种改进的用户最近邻选择方法, 当用户数量较大时, 可以比较快速、准确地找到目标用户的最近邻居, 在一定程度上可以改善扩展性问题。

## 2 基于资源的协作过滤推荐算法

基于目前国内外研究现状, 提出了一种新的协作过滤算法, 算法描述如下:

算法 1 基于资源的协作过滤推荐算法(ICCF)

输入: 测试数据集

步骤 1 对资源项目依内容进行分类;

步骤 2 将用户—资源项目评分矩阵转换为用户—资源类别评分矩阵;

步骤 3 对用户—项目类别评分矩阵进行加权数据预处理;

步骤 4 用户聚类;

步骤 5 计算目标用户与其它用户的相似性;

步骤 6 选取目标用户的若干最相似用户(最近邻居), 建立最近邻居集;

输出: 目标用户的推荐集。

算法各步骤具体描述如下。

### 2.1 对资源项目分类

根据资源项目具体的内容, 将其进行分类。分类的方法有很多, 基于传统技术的分类方法主要包括支持向量机(SVM)、 $K$  近邻(KNN)、决策树、基于关联规则的、基于数据库的和贝叶斯分类等算法, 当然也可以利用资源已有的分类体系来完成对资源的分类。

### 2.2 建立用户—资源类别评分矩阵

随着资源项目的日益增多, 用户的评分数据却一般只集中在自己感兴趣的领域, 前者的增长速度远远高于后者, 这样数据的稀疏性就日益突现。虽然资源项目越来越多, 但其可以根据自身的内容, 划分为不同的类别, 而且类别总数还相对比较固定。故而可将用户—资源项目评分矩阵转换为用户—资源类别评分矩阵, 这样就可以大大降低数据的维度和稀疏程度。由于每项资源依据其内容划分可能属于多个不同的类别, 在进行矩阵转换时, 可将用户对每项资源的评分转换为用户对其所属类别的平均评分。假设资源项目被分成  $X$  个类别, 则可以将用户—资源项目评分矩阵  $R(M \times N)$  转化为用户—资源类别评分矩阵  $C(M \times X)$ 。矩阵  $C(M \times X)$  如表 2 所示, 其中每项  $C_{m,x}$  表示用户  $m$  对资源项目类别  $x$  的评分。

### 2.3 数据预处理

考虑到每个用户评分数据的多少以及频率等方面的影响,

表2 用户—资源项目类别评分矩阵  $C(M \times X)$ 

User	Category				
	Category <sub>1</sub>	...	Category <sub>j</sub>	...	Category <sub>x</sub>
User <sub>1</sub>	$C_{1,1}$	...	$C_{1,j}$	...	$C_{1,x}$
...	...	...	...	...	...
User <sub>i</sub>	$C_{i,1}$	...	$C_{i,j}$	...	$C_{i,x}$
...	...	...	...	...	...
User <sub>M</sub>	$C_{M,1}$	...	$C_{M,j}$	...	$C_{M,x}$

采用文献[26]中提出的一种加权过滤(Weighted Sifting)的方法,对  $C(M \times X)$  中的数据进行预处理。依次运用如下公式(1)~(4)得到最终的矩阵  $CW(M \times X)$ 。

$$cf_{i,j} = \frac{c_{i,j}}{\max[c_{i,1}, c_{i,2}, \dots, c_{i,n}]} \quad (1)$$

$$ipf_i = \left( \log \frac{X}{x_i} \right) \quad (2)$$

$$w_{i,j} = cf_{i,j} \times ipf_i \quad (3)$$

$$CW_{i,j} = C_{i,j} + W_{i,j} \quad (4)$$

公式(1)中  $\max[c_{i,1}, c_{i,2}, \dots, c_{i,n}]$  表示用户  $i$  对所有资源类别的最大评分值;公式(2)中的  $X$  代表项目类别的总数,  $x_i$  代表用户  $i$  所评价的不同类别的个数。

## 2.4 用户聚类

在  $CW(M \times X)$  矩阵的基础上,采用  $K$ -平均聚类算法对用户进行聚类。聚类可以离线进行,能满足推荐系统适时性的要求。

## 2.5 用户间相似性度量和最近邻选择

在基于用户的协作过滤算法中,度量两个用户之间的相似性,一般有两种方法:余弦相似性、相关相似性。

(1)余弦相似性:把用户评分看作是  $N$  维资源空间上的向量。如果用户对某项资源没有评分,则将此评分假设为 0。通过计算两个向量之间的夹角余弦来度量两个用户之间的相似性,计算公式如式(5)所示:

$$sim(i, j) = \frac{\sum_{n=1}^N R_{i,n} * R_{j,n}}{\sqrt{\sum_{n=1}^N R_{i,n}^2 * \sum_{n=1}^N R_{j,n}^2}} \quad (5)$$

其中  $R_{i,n}$  和  $R_{j,n}$  分别表示用户  $i$  和  $j$  对资源  $n$  的评分,  $N$  代表资源项目的总数。

(2)相关相似性(Pearson 相关性):通过 Pearson 相关系数来度量两个用户的相似性。计算时,首先找到两个用户共同评分过的项目集,然后计算这两个向量的相关系数,计算公式如(6)所示:

$$sim(i, j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i) * (R_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I_{ij}} (R_{i,c} - \bar{R}_i)^2 * \sum_{c \in I_{ij}} (R_{j,c} - \bar{R}_j)^2}} \quad (6)$$

其中  $I_{ij}$  表示用户  $i$  和  $j$  共同评分过的项目集,  $R_{i,c}$ 、 $R_{j,c}$  表示用户  $i$  和  $j$  对  $I_{ij}$  中某一项资源的评分,  $\bar{R}_i$ 、 $\bar{R}_j$  表示用户  $i$  和  $j$  对资源的平均评分。算法实现时可以选择其中一种来计算用户间的相似性。

传统协作过滤推荐算法(CF)寻找目标用户最近邻居在整个用户—项目矩阵中来寻找,工作量很大,而且无法避免或减少数据稀疏性所带来的计算最近邻不准确的问题。而本文算法

在各个用户聚类中计算目标用户和其他用户的相似性,再选取前若干个最相似的用户作为其最近邻居,这样不仅可以缩小目标用户最近邻居的查找范围,而且也增加了最近邻居选择的准确性,提高了协作过滤推荐算法的扩展性。

## 2.6 信息推荐

计算目标用户  $u$  对未评分项目  $i$  的评分时,根据最近邻居对项目  $i$  的评分产生推荐,采用公式(7)计算目标用户  $u$  对资源项目  $i$  的预测评分:

$$P_{u,i} = R_u + \frac{\sum_{a=1}^b (R_{a,i} - \bar{R}_a) * sim(u, a)}{\sum_{a=1}^b sim(u, a)} \quad (7)$$

其中  $\bar{R}_u$ 、 $\bar{R}_a$  分别表示用户  $u$  和其最近邻  $a$  对资源项目的平均评分,  $R_{a,i}$  表示最近邻  $a$  对项目  $i$  的评分,  $b$  为最近邻的个数,  $sim(u, a)$  表示用户  $u$  和  $a$  的相似度。

## 3 仿真实验及结果分析

### 3.1 实验数据

实验采用的数据集是协作过滤推荐算法常采用的标准测试数据集——MovieLens (<http://www.grouplens.org>),其中包括 943 位用户对 1 682 部电影的 10 万条评分记录以及电影所属类别等详细信息。在数据集中,每个用户至少有 20 条评分记录,评分值为 1~5 的整数,数值越高,表明用户对该电影的喜爱程度越高,反之则表示用户对其喜爱的程度越低。

实验中依据数据集资源已有的分类信息把 1 682 部电影分成 18 个类别,分别是 Action、Adventure、Animation、Children's、Comedy、Crime、Documentary、Drama、Fantasy、Film-Noir、Horror、Musical、Mystery、Romance、Sci-Fi、Thriller、War、Western。这样,整个用户项目评分数据就可以映射到这 18 维数据空间上。由于数据集中每一部电影可能属于多个类别,因此可以将用户对资源的评分转化为用户对资源类别的平均评分。比如用户  $a$  对电影 1、2 的评分分别为 5 分和 3 分,而电影 1 属于类别 Action 和 Adventure,电影 2 属于类别 Action 和 War,那么用户 1 对 Action 类别的评分可计算为  $(5+3)/2=4$  分。

### 3.2 数据稀疏度实验

数据的稀疏度常用评分数据矩阵中未评分项目所占的百分比来表示。MovieLens 原数据集的稀疏度为:  $1-100\ 000/(1\ 682*943)=0.936\ 953$ ;而将此数据集的用户评分映射到这 18 个类别上后,矩阵中非 0 的评分数据有 14 672 条,所以用户—资源项目类别矩阵的数据稀疏度为:  $1-14\ 672/(943*18)=0.135\ 619$ ,由此可见数据的稀疏程度得到了大大的降低。

### 3.3 扩展性实验

以用户对资源项目类别的评分数据为基础,分别运用  $K$ -平均聚类算法对用户进行聚类,分成 5 和 10 个类别。两种聚类各簇中用户的个数占总用户数的百分比分别如图 1、图 2 所示。

传统协作过滤算法计算用户间的相似性、寻找目标用户最近邻居都在整个用户资源评分数据集上进行,而本章提出的算法中,上述两项工作都只在目标用户自己所在的簇中进行,从图 1 和图 2 可以看出,由于各簇中用户数明显减少,致使这两项工作的计算量显著减少,从而有效地改善了因用户数目过

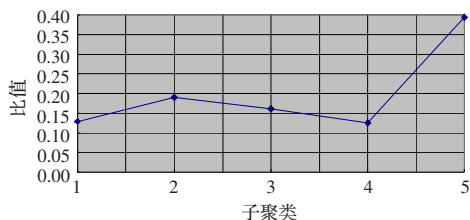


图1 聚类数为5的各簇中用户所占的比例

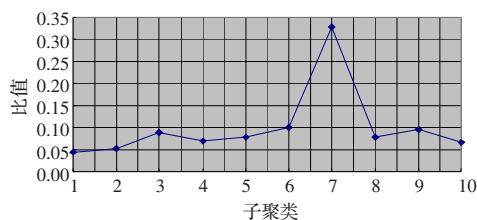


图2 聚类数为10的各簇中用户所占的比例

表3 在 ML\_1 上各算法 MAE 值对比

算法	MAE						
	最近邻数						
	5	10	15	20	25	30	35
传统的协作过滤算法	1.580 374	1.622 231	1.650 303	1.679 378	1.701 827	1.729 737	1.750 050
ICCF_5	0.828 556	0.818 498	0.816 228	0.814 647	0.814 727	0.814 954	0.815 032
ICCF_10	0.828 769	0.819 532	0.816 697	0.814 664	0.815 058	0.815 371	0.815 228

表4 在 ML\_2 上各算法 MAE 值对比

算法	MAE						
	最近邻数						
	5	10	15	20	25	30	35
传统的协作过滤算法	1.569 066	1.612 208	1.645 840	1.681 166	1.710 020	1.736 423	1.759 550
ICCF_5	0.792 264	0.782 201	0.778 189	0.776 606	0.777 041	0.777 456	0.777 182
ICCF_10	0.791 718	0.782 424	0.778 992	0.777 713	0.777 842	0.777 763	0.777 820

表5 在 ML\_3 上各算法 MAE 值对比

算法	MAE						
	最近邻数						
	5	10	15	20	25	30	35
传统的协作过滤算法	1.579 599	1.623 844	1.661 227	1.700 047	1.727 017	1.748 408	1.773 571
ICCF_5	0.785 167	0.773 210	0.769 665	0.768 099	0.767 615	0.767 323	0.767 003
ICCF_10	0.785 988	0.773 578	0.770 156	0.768 694	0.768 724	0.768 356	0.768 737

表6 在 ML\_4 上各算法 MAE 值对比

算法	MAE						
	最近邻数						
	5	10	15	20	25	30	35
传统的协作过滤算法	1.553 347	1.584 602	1.625 078	1.653 400	1.676 561	1.697 959	1.720 350
ICCF_5	0.807 070	0.796 432	0.792 086	0.791 085	0.789 837	0.788 589	0.787 913
ICCF_10	0.807 673	0.796 408	0.793 292	0.792 026	0.791 003	0.790 697	0.790 483

表7 在 ML\_5 上各算法 MAE 值对比

算法	MAE						
	最近邻数						
	5	10	15	20	25	30	35
传统的协作过滤算法	1.545 493	1.578 098	1.607 112	1.640 453	1.667 464	1.686 927	1.709 975
ICCF_5	0.781 947	0.772 825	0.770 768	0.768 480	0.766 933	0.767 111	0.767 379
ICCF_10	0.781 953	0.772 327	0.770 148	0.768 390	0.767 376	0.767 783	0.768 060

多等原因造成的协作过滤算法所面临的扩展性问题,而且聚类可以离线进行,满足了协作过滤算法适时性的要求。

### 3.4 预测精度实验

平均绝对偏差 MAE (Mean Absolute Error) 是最常用的一种推荐质量度量方法。通过计算预测的用户评分与实际用户评分之间的偏差来度量预测的准确性, MAE 越小, 推荐质量就越高。MAE 按公式(8)计算:

$$MAE = \frac{\sum_{u \in T} |R_u(t_j) - P_u(t_j)|}{|T|} \quad (8)$$

上式中  $T$  表示测试集合,  $|T|$  表示测试集中被预测评分项目的总数,  $R_u(t_j)$  和  $P_u(t_j)$  分别表示用户  $u$  对项目  $t_j$  的实际评分和预测评分。实验中采用平均绝对偏差 MAE 作为算法推荐质量的度量标准。

为方便做交叉实验, 将评分数据集平均分成 5 等份, 依次表示为 ML\_1, ML\_2, ML\_3, ML\_4, ML\_5, 每次分别选取其中一份作为测试集, 剩下的数据作为训练集。依次选取测试集中目标用户的前 5, 10, 15, 20, 25, 30, 35 个最近邻居, 将传统的协作过滤算法 (CF) 和本章提出的基于资源内容划分的协作过滤算法 (ICCF) 两者的 MAE 值作了比较, 实验结果分别如表 3~表 7 所示。其中 ICCF\_5、ICCF\_10 分别表示运用  $K$ -平均聚类算法

对用户进行聚类,分成5和10个类别的ICCF算法,本实验中用户间相似性度量皆选取余弦相似性来计算。

从表3~表7交叉实验的结果可以看出,不管是将用户聚类成5个还是10个类别,在不同的测试集上,本文提出的推荐算法ICCF在推荐质量上都要优于传统的协作过滤算法。同时从部分MAE曲线可以看出,当最近邻数目达到一定程度以后,MAE值呈增长趋势,即推荐质量下降。原因是当最近邻数目较多时,相似系数过低的用户也被选入最近邻居集而参加推荐,这样势必造成推荐质量下降。所以为了保证推荐质量,应选取一定数目的用户或相似性达到一定阈值的用户作为最近邻。在不同的测试集上,ICCF\_5的MAE值基本上均要略低于ICCF\_10的MAE值,即ICCF\_5的推荐精度要高于ICCF\_10。由此说明,ICCF算法在对用户进行不同的聚类时,也影响着算法推荐质量的好坏。所以研究并选取一种更加有效的聚类算法,可以提高推荐算法的信息推荐质量。

#### 4 总结和展望

本文提出了一种基于资源内容划分的协作过滤推荐算法,在对资源项目依内容划分的基础上,将用户—项目评分数据转换为用户—资源类别评分数据,在对此评分数据运用加权过滤方法预处理之后,对用户进行聚类,在目标用户所在的簇中寻找其最近邻居并产生推荐。实验结果显示,该算法较传统的协作过滤推荐算法,不仅大大降低了数据的稀疏程度,而且明显提高了算法的扩展性和信息推荐的质量。

#### 参考文献:

- [1] 曾春,刑春晓,周立柱.个性化服务技术综述[J].软件学报,2002,13(10):1952-1961.
- [2] Breese J, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]//Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, 1998:43-52.
- [3] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International WWW Conference. New York: ACM Press, 2001:285-295.
- [4] Chickering D, Heckerman D. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables[J]. Machine Learning, 1997, 29(2):181-212.
- [5] Sarwar B, Karypis G, Konstan J, et al. Analysis of recommendation algorithms for E-commerce[C]//Proceedings of ACM Conference on Electronic Commerce, 2000:158-167.
- [6] Wolf J, Horting A C. Hatches an egg: a new graph-theoretic approach to collaborative filtering[C]//Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, San Diego, 1999:201-212.
- [7] 邓爱林,朱扬勇,施伯乐.基于项目预测评分的协同过滤推荐算法[J].软件学报,2002,13(4):1-8.
- [8] 周军锋,汤显,郭景峰.一种优化的协同过滤推荐算法[J].计算机研究与发展,2004,41(10):1842-1847.
- [9] Deng Ai-lin. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of Software, 2003, 14(9):1621-1628.
- [10] Billsus D, Pazzani M. J. Learning collaborative information filtering[C]//Proceedings of the International Conference on Machine Learning, Madison, 1998:46-54.
- [11] Arnd K, Bernard M. Clustering for collaborative filtering applications[C]//Proceedings of Computational Intelligence for Modeling, Control and Automation. Vienna, Austria: IOS Press, 1999:17-19.
- [12] 程岩,肖小云,吴洁倩.基于聚类分析的电子商务推荐系统[J].计算机工程与应用,2005,41(24):175-177.
- [13] Ansari A, Essegaier S, Kohli R. Internet recommendation systems[J]. Journal of Marketing Research, 2000, 37:363-375.
- [14] Sarwar B, Konstan J A. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system[C]//Proc ACM Conf Computer Supported Cooperative Work (CSCW). New York: ACM Press, 1998:345-354.
- [15] Good N, Schafer J B, Konstan J A. Combining collaborative filtering with personal agents for better recommendations[C]//Conference of the American Association of Artificial Intelligence. New York: AAAI Press, 1999:439-446.
- [16] 张巍,刘鲁,葛健.一种基于粗集的协同过滤算法[J].小型微型计算机系统,2005,26(11):1972-1974.
- [17] Mobasher B, Jin X, Zhou Y. Semantically enhanced collaborative filtering on the Web[C]//Proceedings of the European Web Mining Forum. Resnik: Springer Press, 2004:57-76.
- [18] Sarwar B, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system—a case study[C]//Proc ACM WebKDD 2000 Web Mining for E-Commerce Workshop. New York: ACM Press, 2000:82-90.
- [19] 赵亮,胡乃静,张守志.个性化推荐算法设计[J].计算机研究与发展,2002,39(8):986-991.
- [20] Rosenstein M, Lochbaum C. Recommending from content: Preliminary results from an e-commerce experiment[C]//Proceedings of Conference on Human Factors in Computing. Heidelberg Berlin: ACM Press, 2000:291-292.
- [21] Honda K, Sugiura N, Ichihashi H. Collaborative filtering using principal component analysis and fuzzy clustering[C]//Proceedings of the 1st Asia-Pacific Conference on Web Intelligence: Research and Development. Maebashi City, Japan: Springer Press, 2001:394-402.
- [22] 高凤荣,杜小勇,王珊.一种基于稀疏矩阵划分的个性化推荐算法[J].微电子学与计算机,2004,21(2):58-62.
- [23] 潘红艳,林鸿飞,赵晶.基于矩阵划分和兴趣方差的协同过滤算法[J].情报学报,2006,25(1):49-54.
- [24] Aggarwal C C. On the effects of dimensionality reduction on high dimensional similarity search[C]//Proceedings of the twentieth ACM PODS Conference, Santa Barbara, California, USA, 2001:256-266.
- [25] 王霞,刘琴.协同过滤在推荐系统中的应用研究[J].计算机系统应用,2004(4):24-27.
- [26] Sumiya T, Chun J, Lee Sang-goo. A weighted sifting method to improve the effectiveness of collaborative filtering[C]//2004 IEEE Region 10 Conference, 2004, 2:266-269.